

第 18 课：多态之抽象类案例

讲师：Monkey

时间：2017 年 5 月 8 日（周一）

课程要点：

1.抽象类编程案例

2.虚方法抽象类语法对比

1.抽象类编程案例

1.前置回顾

<1>关于多态的实现方式已经介绍了虚方法，抽象类两种方式了。

<2>多态的使用前提，是建立在继承的关系之上的，也就是说必须要先有继承关系，然后才会出现多态。

<3>面向对象的封装，继承，多态，都是我们后期规划代码结构的基本思想。

<4>后续使用 Unity 开发一款游戏，大点的项目可能会有几百个独立的脚本文件，这么多的脚本文件，如果没有一个代码结构框架来管理的话，项目十有八九是会中途夭折的。

2.使用抽象类结构实现 NPC 模块

在游戏中会出现很多种不同用途的 NPC，这些 NPC 有各自的存在价值和作用，同时又具备一些共性的东西。在开发 NPC 系统的时候，往往是需要提取共性，独立出一个父类，然后子类继承实现不同作用的 NPC。

分析：

任务 NPC，商贩 NPC，铁匠 NPC，三种 NPC 的种类。

共有属性：npc 的名字，npc 的类型；

共有方法：都能和玩家交互(交谈)；

2.虚方法抽象类语法对比

对比分析下虚方法和抽象类（抽象方法）的语法格式 [见图]

重点记忆语法格式，特有的关键字。

如果是刚接触编程的新人，不要指望学完了马上就会用，先掌握语法结构，看别的人代码看到这关键字后，知道哪个是抽象类，哪个是虚方法，就够了。